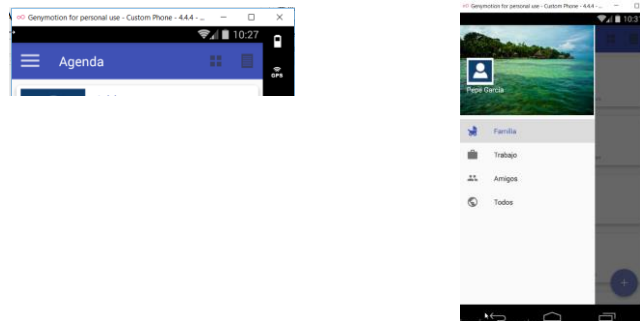


Menú lateral

Modifica la agenda de manera que en la aplicación principal, al pulsar sobre el icono Home o al hacer deslizamiento, mostraremos un menú lateral que nos dará la opción de seleccionar que se muestren todos los contactos, o solo los filtrados por algún grupo.



Pasos a seguir para realizar el Filtrado correctamente

Para aplicar filtros a nuestros datos el adaptador debe implementar la interface `Filterable`, concretamente sobrescribiremos el método `getFilter()`. Este método se encargará de implementar el filtro sobre nuestros datos a través de un objeto de tipo `Filter` (incluido en el adaptador). Esta clase contiene dos métodos abstractos `performFiltering()` y `publishResults()`. El primero realiza el filtro de los datos y el segundo notifica los cambios.

Veamos en primer lugar como quedaría la parte afectada del adaptador:

```
class AdaptadorContacto extends RecyclerView.Adapter implements View.OnClickListener,
    View.OnLongClickListener, View.OnTouchListener, OnClickListenersImagen, Filterable {
    private ArrayList<Contacto> datos, datosOriginales;
    private View.OnClickListener listener;
    private View.OnLongClickListener longListener;
    private View.OnTouchListener listenerTouch;
    private OnClickListenersImagen listenerImagen;
    private Integer orientacion;

    private Filter filterable;

    ...

    @Override
    public Filter getFilter() {
        if(filterable==null) filterable=new ContactosFilter();
        return filterable;
    }

    private class ContactosFilter extends Filter {

        @Override
        protected FilterResults performFiltering(CharSequence seleccion) {
            FilterResults results=new FilterResults();
            ArrayList<Contacto> datosFiltrados=new ArrayList<>();
            if(seleccion==null || seleccion.length()==0) datosFiltrados=datosOriginales;
            else if(seleccion.equals("Todo")) datosFiltrados=datosOriginales;
            else
            {
                for(Contacto c: datosOriginales)
                {
                    if(seleccion.equals("Amigo") && c.amigos) datosFiltrados.add(c);
                    else if(seleccion.equals("Familiar") && c.familiar) datosFiltrados.add(c);
                    else if(seleccion.equals("Trabajo") && c.trabajo) datosFiltrados.add(c);
                }
            }
            results.values=datosFiltrados;
            results.count=datosFiltrados.size();
            return results;
        }

        @Override
        protected void publishResults(CharSequence charSequence, FilterResults filterResults) {
            datos=(ArrayList<Contacto>)filterResults.values;
            notifyDataSetChanged();
        }
    }
}
```

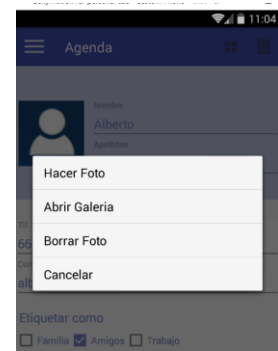
Ejercicio Propuesto Agenda Menús

Para aplicar el filtro cuando se pulsa sobre cada una de las opciones del menú, deberemos implementar el código en el método que escucha el ítem pulsado del menú lateral (setNavigationItemSelectedListener). Podría ser algo parecido a lo siguiente:

```
void accionMenuLateral(int id)
{
    switch (id)
    {
        case R.id.navigation_item_familia:
            fragmentRecycler.adapter.getFilter().filter("Familiar");
            break;
        case R.id.navigation_item_amigo:
            fragmentRecycler.adapter.getFilter().filter("Amigo");
            break;
        case R.id.navigation_item_trabajo:
            fragmentRecycler.adapter.getFilter().filter("Trabajo");
            break;
        case R.id.navigation_item_todo:
            fragmentRecycler.adapter.getFilter().filter(null);
            break;
    }
}
```

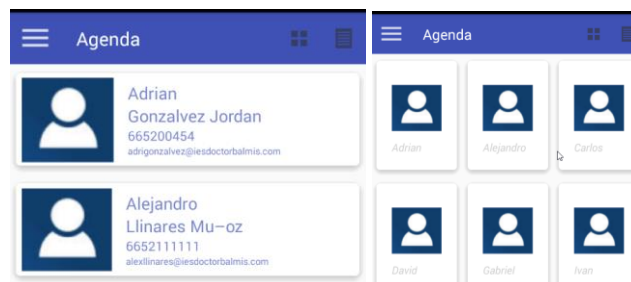
Menú emergente

Si pulsamos en la foto **tanto del editar como del añadir**, se lanzará un menú emergente, como se muestra en la imagen, que nos permitirá eliminar, abrir galería o hacer foto. Una vez se haya seleccionado una foto, deberá mostrarse como la imagen de ese usuario en la lista.



Menú de opciones

Usaremos el menú de opciones de la ActivityMain para darnos la oportunidad de mostrar la vista en modo parrilla o modo lista.



Cuando la aplicación arranque, detectará si el dispositivo está en horizontal o vertical para cargar la vista correspondiente, parrilla o lista respectivamente. Pero una vez la aplicación esté arrancada el usuario podrá seleccionar cualquier modo de vista, independientemente de la posición en la que se encuentre el dispositivo.

Para controlar si la aplicación está en horizontal o vertical podremos usar lo siguiente:

If(getResources().getConfiguration().orientation==Configuration.ORIENTATION_PORTRAIT)

En cuanto a la eliminación, llamada o mensaje se quedará como ya habíamos implementado en la versión anterior, para ambas vistas

Ejercicio Propuesto Agenda Menús